# Automated Planning for Interferometer Configuration and Control[1]

Gregg Rabideau, Leonard Reder, Steve Chien, Andrew Booth
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91108
818-393-5364
gregg.rabideau@jpl.nasa.gov

*Abstract*—In this paper, we discuss Artificial Intelligence (AI) planning and scheduling technology and how it can be applied to interferometer configuration and control. Interferometers are a scarce resource for the scientific community. Therefore, there is an incentive for streamlining and optimizing the operation of these instruments. Moreover, harsh operating environments make manual operation impractical, further motivating the use of automation. We use the ASPEN automated planning system developed at JPL to demonstrate how planning can be used to perform many operations tasks with many benefits over manual operations. Automated planners can produce large command sequences in a short period of time. In addition, repairing altered plans is very fast and can be used to automatically adapt to situations not originally planned for.

TABLE OF CONTENTS



## 1. INTRODUCTION

Interferometers have received much attention recently because of their ability to perform very high-resolution measurements. In particular, they have been a valuable tool in detecting planets orbiting distant stars. Interferometers, however, are typically complex systems with many subsystems. They involve a great deal of complex closed loop control with intricate sequencing required to control the instrument as a whole. In optical interferometry, the light from two telescopes observing the same object is combined to produce an interference fringe. The light must travel down long "beam trains" through a complex series of mirrors which direct it down optical delay lines and finally into a beam combining optic to produce fringe patterns. All of this is used to obtain high angular resolution information.

Controlling and configuring the hardware for such experiments can be a difficult task. The telescopes and associated optics must be controlled and synchronized with very high precision. Automation is a requirement for certain tasks, such as closed-loop tracking for maintaining the interference fringes. Moreover, to meet the high precision requirements, interferometers must be operated in locations that do not suffer from high atmospheric disturbances. While space and high altitude environments remove or reduce atmospheric disturbances, they do not provide desirable conditions for manual operations. This is one of the driving factors for using additional automation on the Keck Interferometer set high atop Mauna Kea in Hawaii. Finally, interferometers are a scarce resource for the scientific community. Therefore, there is an incentive for streamlining and optimizing the operation of these instruments.

Planning and sequencing software can be used to automate configuration and control tasks at many levels. High-level sequencing is required to coordinate the subsystems and schedule sets of experiments. Low-level sequencing is required to implement controllers that perform such tasks as mode switching within fringe and angle tracking subsystems and alignment. AI planning addresses more complex problems, but typically must employ less efficient algorithms. Event driven sequencers can be fast, but usually do not address more detailed timing and resource issues.

Given a high-level science request, an AI planner can generate command sequences that control and coordinate subsystems in order to perform the request. For example, a scientist may request a particular type of observation of a star on a given night. From this, the planner can generate commands for pointing the two telescopes at the star, configuring the subsystems for the type of observation, aligning mirrors along the beam train, etc. In addition, the planner can consider other observations during the same night, and schedule them at non-conflicting times. This requires managing the resources used by each observation and detecting over-subscription. The planner can consider various other constraints, such as operational ordering

[1] 0-7803-6599-2/01/$10.00 © 2001 IEEE

constraints between commands. For example, in some cases a target star must be in a certain position within the aperture of the telescope before fringes can be acquired. Automated planners can produce large command sequences in a short period of time. In addition, repairing altered plans is very fast and can be used to adapt to situations not originally planned for.

Each command generated by the planner may require additional low-level sequencing. In particular, many closed-loop control tasks must be highly optimized. Event-driven sequencing is more suited for problems with tighter real-time requirements. For problems with fixed state sets and well-defined state transitions, sequential operations can be programmed and executed based on state transition diagrams. For example, a single mirror alignment command may actually start a sequence of commands that repeatedly computes the centroid of a reflected light beam and moves the mirror to change the centroid location. Event-driven sequencers can move through state transitions until the desired state (e.g., "aligned") is met.

In this paper, we focus on using automated planning and scheduling technology for high-level interferometer configuration and coordination. We use the ASPEN system developed at JPL to demonstrate how planning can be used to perform many operations tasks with many benefits over manual operations. We also investigate the integration of this technology with sequencing for low-level control using the EPICS framework. Specifically, our demonstration uses specifications and testbed software for the Keck Interferometer. In the next section, we describe Keck, its operational constraints and other aspects relevant to planning. Then, we describe the use of low-level subsystem sequencers. Then, we describe a prototype for high-level planning with the ASPEN system. After that, we describe extensions for continuous planning and for optimization. Finally, we discuss conclusions and future work.

## 2. THE KECK INTERFEROMETER

The Keck Interferometer (KI) [1] is funded by NASA as a joint development between the Jet Propulsion Laboratory, California Institute of Technology (JPL) and the W. M. Keck Observatory, California Association for Research in Astronomy (CARA). Located on Mauna Kea in Hawaii (see Figure 1), KI is a ground-based component of NASA's Origins Program. Origins addresses fundamental questions about the formation of galaxies, stars, and planetary systems, the prevalence of planetary systems around other stars, and the formation of life on Earth. The Keck Interferometer will combine the two existing 10-m Keck telescopes with four proposed 1.8-m outrigger telescopes as an interferometric array capable of addressing a broad range of astronomical science.

The Keck Interferometer will use Michelson beam combination among the two Kecks and the four outriggers. The two Kecks provide a baseline of 85 m, while the

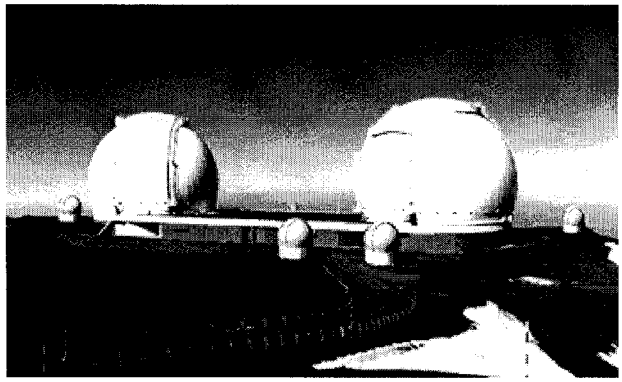


Figure 1 - An artist's rendition of the final Keck Interferometer configuration on Mauna Kea in Hawaii.

baselines available with the outriggers will be between 25 m and 140 m. The interferometer will combine phased pupils provided by adaptive optics on the Kecks and fast tip/tilt correction on the outriggers. Cophasing of the array will be accomplished by fringe tracking on an isoplanatic reference to enable high-sensitivity science observations. Key components of the cophasing system include active delay lines in the beam-combining lab and dual-star modules at each telescope. Several back-end beam combiners will be provided, including two-way beam combiners at 1.5–2.4 μm for fringe tracking, astrometry, and imaging; a multi-way combiner at 1.5–5 μm and 10 μm for imaging; and a nulling combiner for high dynamic range observations at 10 μm.

The design of the interferometer and its instrumentation is responsive to several key Origins science objectives. Science programs with the interferometer using the two Kecks include:

- *Characterization of exozodiacal dust*
  The quantity of exozodiacal dust around other solar systems is poorly known, especially down to levels near that of our solar system. The exozodiacal dust is a noise source for future space imaging missions. The Keck Interferometer will combine the two 10-m Keck telescopes using interferometric nulling to make this measurement down to levels less than ten times our solar system.
- *Direct detection of hot Jupiters and brown dwarfs*
  Because of the different spectra of these objects compared with the stars they orbit, the center of light of the star-planet system is wavelength dependent and can be sensed with multi-wavelength phase measurements. Direct measurements allow detections in a single night.

Science programs incorporating the outriggers include:

- *Astrometric detection of exoplanets*

By sensing the reflex motion of a star caused by an orbiting planet, the Keck Interferometer will be able to survey hundreds of nearby stars for planets to Uranus mass. This program uses the outriggers alone to implement high accuracy narrow-angle astrometry.

- *Six-way interferometric imaging*
  The interferometer can be configured for imaging using the 4 outriggers only, 4 outriggers with one Keck, and 4 outriggers with both Kecks. The potential imaging science includes the observation of protoplanetary disks, to detect evidence of planetary formation, as well as a variety of other Origins and astrophysical targets.

## 3. SEQUENCING FOR KI

The current plan for Keck Interferometer Sequencing is to use EPICS Sequencers for all subsystems. The Experimental Physics and Industrial Control System (EPICS) [2] is a distributed process control system that provides data acquisition, supervisory control, closed-loop control, archiving, and alarm management. Sequential operations for real-time control can be easily programmed using the State Notation Language (SNL) [3]. SNL programs are based on concepts from state transition diagrams and are executed using the EPICS Sequencer. KI sequencers will be running on both Unix and embedded VxWorks systems.

Within the Keck Interferometer there are several subsystems controlled by EPICS sequencing. These include the fringe tracker (FT), the fast atmospheric tracking camera (FATCAT), long and fast optical delay lines (LDLs and FDLs), the angle tracker (KAT), and an automatic alignment system. The most critical sequencing for the interferometer is the sequencing required for obtaining fringe tracker lock on a stellar object. This sequencer must orchestrate all of the above sub-systems. Currently, most of the sequencing for KI is still in the definition stage.

Before each night of observations, the interferometer must be aligned. The automatic alignment system consists of two types of alignment sequencers and a top-level executive control program. The initial approach is to have one instance of an EPICS state notation sequencer aligning each optic in the system. Each sequencer will be either an Internal Instrument Alignment Sequencer (IIAS) or a Beam Train Alignment Sequencer (BTAS). The IIAS is used to optimize the light passing into both the FATCAT and KAT subsystems while the BTAS is utilized for alignment of all mirrors in the beam train from the telescopes to the beam combining optics.

For mirror to mirror alignment we use the Beam Train Alignment Sequencer (BTAS). The BTAS state transition diagram can be seen in Figure 2 (the details are beyond the

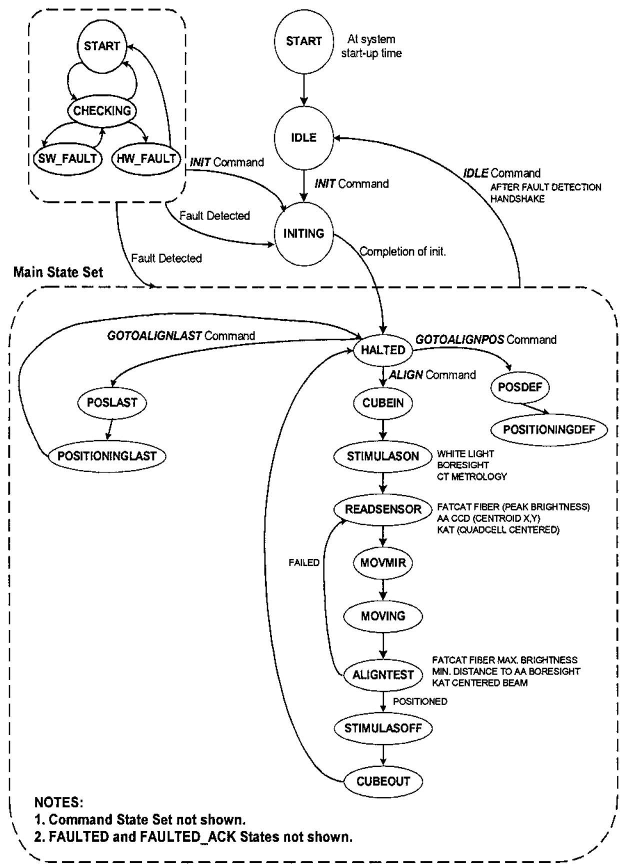



Figure 2 - The Beam Train Alignment Sequencer.

scope of this paper). The sequencer turns on and off LED alignment targets downstream of each optic and than acquires centroid information from a CCD camera and framegrabber using specialized EPICS records. Based on the calculated centroid, the sequencer then sends a move command to the mirror. The sequencer repeats these steps until alignment is achieved. A separate sequencer handles this task for each mirror working from the end of the beam train back upstream to the telescope. This alignment establishes an optical path for the target light from the telescope to appropriate combiner, where it is combined with light from one or more of the other telescopes.

EPICS sequencers execute state transition diagrams implemented in a simple yet powerful language. These fast, event-driven sequencers are well suited for real-time problems with fixed state sets and well-defined state transitions. In contrast, AI planning and scheduling generates the set of activities that satisfy a set of high-level goals. Appropriate state transitions are made while considering multiple alternatives, resources, and timing constraints. AI planning addresses more complex problems, but typically must employ less efficient algorithms.

## 4. The ASPEN Planner

We show that AI planning and scheduling technology can be used to solve the interferometer configuration and control problem. ASPEN (Automated Scheduling and Planning ENvironment) encodes system operability constraints, rules, hardware models, science experiment goals, and operations procedures to allow for automated generation of low-level sequences [4]. By automating the command sequence generation process and by encapsulating the operations specific knowledge, ASPEN enables large systems to be controlled by a small operations team–thereby reducing costs.

ASPEN is a planning and scheduling software framework [5] that provides a reusable set of software components that implement the elements commonly found in complex planning/scheduling systems. The main components include:

- a language compiler for acquiring constraints and preferences expressed by the user
- generic search algorithms for generating, repairing, and optimizing plans
- a soft, real-time replanning capability
- a Graphical User Interface (GUI) for viewing and modifying plans

The job of a planner/scheduler, whether manual or automated, is to accept high-level goals and generate a set of low-level activities that satisfy the goals, do not violate any of the system rules or constraints, and optimize the quality of the plan. ASPEN has been used on similar configuration and control problems for Deep Space Network (DSN) operations [6].

### *Model Components and Constraints*

System models are developed in the ASPEN Modeling Language (AML) [7]. These models are parsed into data structures that provide efficient reasoning capabilities for planning and scheduling. The seven basic ASPEN model components are: activities, parameters, parameter dependencies, temporal constraints, reservations, resources and state variables. These components are used to describe what the system can and cannot do during operations.

An activity is an occurrence over a time interval that in some way affects the system state. It can represent anything from a high-level goal or request to a low-level event or command. An activity has a set of parameters, parameter dependencies, temporal constraints, reservations and decompositions. All activities have at least three parameters: a start time, an end time, and duration. There is also at least one parameter dependency, relating these three parameters. In addition, all activities have at least one temporal constraint that prevents the activity from occurring outside of the planning horizon. Any additional components are optional.

A parameter is simply a variable with a restricted domain. A parameter dependency is a functional relationship between two parameters. An activity end time, for example, is a function (the sum) of the start time and the duration. A more complicated dependency might compute the duration of a telescope target lock from the initial and final positions.

In the model, relative ordering constraints can be specified for pairs of activities. A temporal constraint is a relationship between the start or end time of one activity with the start or end time of another activity with minimum and maximum separation distances. One might specify, for example, that an initialization activity must end at least one second but at most five minutes before the start of an activity that uses the instrument. Temporal constraints can be combined with conjunctive or disjunctive operators to form more complicated expressions.

A resource represents the profile of a physical resource or system variable over time, as well as the upper and lower bounds of the profile. In ASPEN, a resource can either be depletable or non-depletable. A depletable resource used by a reservation remains used even after the end of the activity making the reservation. Examples of depletable resources include memory and battery energy. A non-depletable resource is used only for the duration of the activity making the reservation. An instrument is an example of a non-depletable resource. A state variable represents the value of a discrete system variable over time and specifies the set of possible states and the set of allowable transitions between states. An example of a state variable is an instrument switch that may be either ON, WARMING, or OFF. This state variable may be restricted to transitions from OFF to WARMING but not directly to ON. The requirements of activities on resources or state variables are called reservations. For example, an activity can have a reservation for ten watts of power.

Activity hierarchies can be specified in the model using decompositions. A decomposition is a set of sub-activities along with temporal constraints between them. In this way, one can define a high-level activity that decomposes into a set of lower-level activities that may be required to occur in some relative order. These activities in turn may have their own decompositions. In addition, an activity may have multiple decompositions to choose from. Thus, allowing an activity to be expanded in different ways.

### *Plan Conflicts and Repair*

We define a conflict as a particular class of ways to violate a plan constraint (e.g., over-use of a resource or an illegal state transition). For each conflict type, there is a set of repair methods. The search space consists of all possible repair methods applied to all possible conflicts in all possible orders. We describe an efficient approach to searching this space.

horizons, each of which lasts for a significant period of time. When one nears the end of the current horizon, one projects what the state will be at the end of the execution of

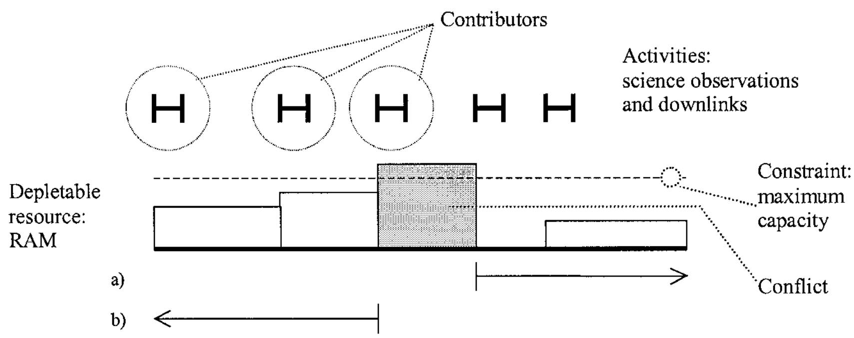



Figure 3 - Repairing a depletable resource conflict. The arrows show time intervals that resolve the conflict by a) moving a positive contributor or b) adding a negative contributor.

In ASPEN, the main algorithm for automated planning and scheduling is based on a technique called iterative repair [8,9]. During iterative repair, the conflicts in the schedule are detected and addressed one at a time until no conflicts exist, or a user-defined time limit has been exceeded. A conflict is a violation of a parameter dependency, temporal or resource constraint. Conflicts can be repaired by means of several predefined methods. The repair methods are: moving an activity, adding a new instance of an activity, deleting an activity, detailing an activity, abstracting an activity, making a reservation of an activity, canceling a reservation, connecting a temporal constraint, disconnecting a constraint, and changing a parameter value. The repair algorithm first selects a conflict to repair then selects a repair method. The type of conflict being resolved determines which methods can repair the conflict. Depending on the selected method, the algorithm may need to make addition decisions. For example, when moving an activity, the algorithm must select a new start time for the activity.

Figure 3 shows an example situation for repair. RAM is represented as a depletable resource. The shaded region shows a conflict where the RAM buffer has been oversubscribed. The science activities using the resource prior to the conflict are considered contributors. Moving or deleting one of the contributors can repair the conflict. Another possibility would be to create a new downlink activity in order to replenish the resource and repair the conflict.

## 5. Continuous Planning

Traditionally, the majority of planning and scheduling research has focused on a batch formulation of the problem. In this approach, when addressing an ongoing planning problem, time is divided up into a number of planning horizons, each of which lasts for a significant period of time. When one nears the end of the current horizon, one projects what the state will be at the end of the execution of the current plan. The planner is invoked with a new set of goals for the new horizon and the expected initial state for the new horizon, and the planner generates a plan for the new horizon. As an example of this approach, the Remote Agent Experiment operated in this fashion [10].

This approach has a number of drawbacks. In this batch oriented mode, typically planning is considered an off-line process which requires considerable computational effort and there is a significant delay from the time the planner is invoked to the time that the planner produces a new plan. If a negative event occurs (e.g., a plan failure), the response time until a new plan is generated may be significant. During this period the system being controlled must be operated appropriately without planner guidance. If a positive event occurs (e.g., a fortuitous opportunity, such as activities finishing early), again the response time may be significant. If the opportunity is short lived, the system must be able to take advantage of such opportunities without a new plan (because of the delay in generating a new plan). Finally, because the planning process may need to be initiated significantly before the end of the current planning horizon, it may be difficult to project what the state will be when the current plan execution is complete. If the projection is wrong the plan may have difficulty.

To achieve a higher level of responsiveness in a dynamic planning situation, we utilize a continuous planning approach and have implemented a system called CASPER (for Continuous Activity Scheduling Planning Execution and Replanning) [11]. Rather than considering planning a batch process in which a planner is presented with goals and an initial state, the planner has a current goal set, a plan, a current state, and a model of the expected future state. At any time, CASPER may generate an incremental update to the goals, current state, or planning horizon (at much smaller time increments than batch planning). This update may be an unexpected event or simply time progressing forward. The planner is then responsible for maintaining a consistent, satisficing plan with the most current information. This current plan and projection is the planner's estimation as to what it expects to happen in the world if things go as expected. However, since things rarely go exactly as expected, the planner stands ready to continually modify the plan.

## 6. PLAN OPTIMIZATION

ASPEN also has facilities for representing and reasoning about plan quality [12]. ASPEN adopts a local, early-commitment, iterative approach to optimization parallel to the iterative repair framework. During iterative optimization, low scoring preferences are detected and addressed individually until the maximum score is attained, or a computational resource limit has been exceeded. A preference is a quality metric for a plan variable, and can be improved by making modifications to the plan similar to repair. For each preference, a domain-independent improvement expert automatically generates modifications that will improve the preference score. For example, minimizing tardiness is a preference on the end time variables of activities and can be improved by moving activities to earlier times.

Experts are local, however, and do not guarantee an increase in overall plan quality. Improvement experts provide a framework for optimization algorithms, defining the search space of possible improvements. We define a separate class of improvement expert for each class of preference. ASPEN currently supports five types of preferences. Local activity variable preferences represent preferences on parameters of an activity such as start time, end time, duration, relative time to other activities, etc. (e.g., minimize separation between science image activity and instrument calibration activity). Activity/goal count preferences are over the number of occurrences of certain types of activities (e.g., maximize science images). Resource/state variable preferences specify desired values for resource levels and state variables (e.g., minimize power usage peak, maximize amount of energy in the battery, keep the buffer as empty as possible). Resource/state change count preferences specify a desire to maximize/minimize the number of times a state or resource changes (e.g., minimize the number of power switches for an instrument). A state duration preference implements a desire to maximize/minimize the amount of time that a given state is true (e.g., minimize instrument on time).

## 7. AN EXPERIMENTAL AI PLANNER FOR KI

We demonstrate AI planning and scheduling technology on a hypothetical operations scenario for a small part of the Keck Interferometer (KI) configuration and control problem. A prototype KI planning system has been developed using the ASPEN framework. This planner could potentially be used for configuration and coordination of the subsystems, allowing the complex Keck system to be centrally controlled at an abstract level. This planning system was integrated with a sequencer currently being developed using the EPICS sequencing tool. When operational, these sequencers will perform closed-loop control of the KI beam train subsystem.

We have adapted the ASPEN framework for use as an automated planning and scheduling system for the Keck Interferometer. This adaptation consists of three parts:

1) developing a constraint model,
2) developing a preference model, and
3) generating a continuous planning interface

*KI Constraint Model*

The first step in any ASPEN application is to develop a model of the system activities and constraints using AML. For KI, we have modeled 9 resources, 37 state variables, 67 activity types. The ASPEN modeling was done with a limited knowledge of the Keck Interferometer and may be different from the actual configuration. However, we believe our prototype demonstrates the potential capabilities of an automated planning system.

We have modeled resources for: each of the two main telescopes, each of the four outriggers, the nulling combiner, the multiway combiner, and the fringe tracker. All are atomic resources, which simply enforce non-parallel usage of each component.

State variables are used to model the pointing of each of the six telescopes, the alignment of each mirror, and the health status of each mirror. The pointing state variables are modeled as a set of discrete states for each of a named set of sky targets. Each target corresponds to a right ascension (RA) and declination (DEC) pair. Each alignment state variable simply models whether or not a mirror is at the appropriate angle for the current target. Finally, each health state variable indicates whether a mirror is working properly or has faulted.

The activity types define a subset of the potential events, commands, and goals that may occur or be requested on the Keck Interferometer. Because many observations must be performed at night, sunrise and sunset are two events that are relevant to Keck operations. Other uncontrollable events include unpredictable situations, such as temporary loss of target from the field of view, and fault situations such as hardware failures.

Some of the KI commands have been modeled as ASPEN activities. These include commands for positioning the telescopes and aligning mirrors along the beam train. A "telescope_search_for_target" activity for positioning a telescope and has parameters that indicate the new target for the telescope. This type of activity reserves the corresponding telescope resource and changes its pointing state variable to the new target state. However, it also changes the alignment state variables (one for each mirror along the beam train of the corresponding telescope) to be "unaligned." This is because the mirrors must be aligned for each new target. There are additional "mirror_alignment" activities for aligning the mirrors when needed. Activities of these types can be scheduled at any time and last for a particular duration.

Finally, we have modeled a set of high-level observation goals for the Keck Interferometer. These goals are eventually detailed into lower level requests for particular resources and states at particular times. At the highest level, we have activities that represent the science experiments (described earlier) for Exozodiacal Dust Measurement, Hot Jupiter Detection, and Astrometric Exoplanet Detection. The Exozodiacal Dust Measurement activity decomposes into interferometry activities that uses the two main telescopes and the nulling combiner. The Hot Jupiter Detection activity also decomposes into interferometry activities using the two main telescopes, but instead has a parameter for indicating which combiner to use. Finally, the Astrometric Exoplanet Detection activity decomposes into astrometry activities that use two of the four outrigger telescopes. Each of the goal activities has parameters for specifying the sky target at which to perform the experiment. Moreover, each goal places requirements on the pointing, mirror alignment, and mirror health of the relevant telescopes for the duration of the experiment.

Once all of the model components are specified, we are now ready to generate observation plans for the Keck Interferometer (see Figure 4). Whether generated manually or automatically, all model constraints will be monitored and conflicts will be flagged. For example, we have automatically generated a three-night plan from an initial request of nine science experiments (three per night). After making the goal requests, conflicts exist from the various goal requirements for pointing, etc. Using the iterative repair algorithm, conflicts were resolved none remained. For example, adding target search activities satisfied the pointing requirement but violated the mirror alignment requirement. The repair algorithm would then add mirror alignment activities to re-align the mirrors after the target search but before the experiment. The algorithm continued this way, finishing with a valid plan of 220 activities.

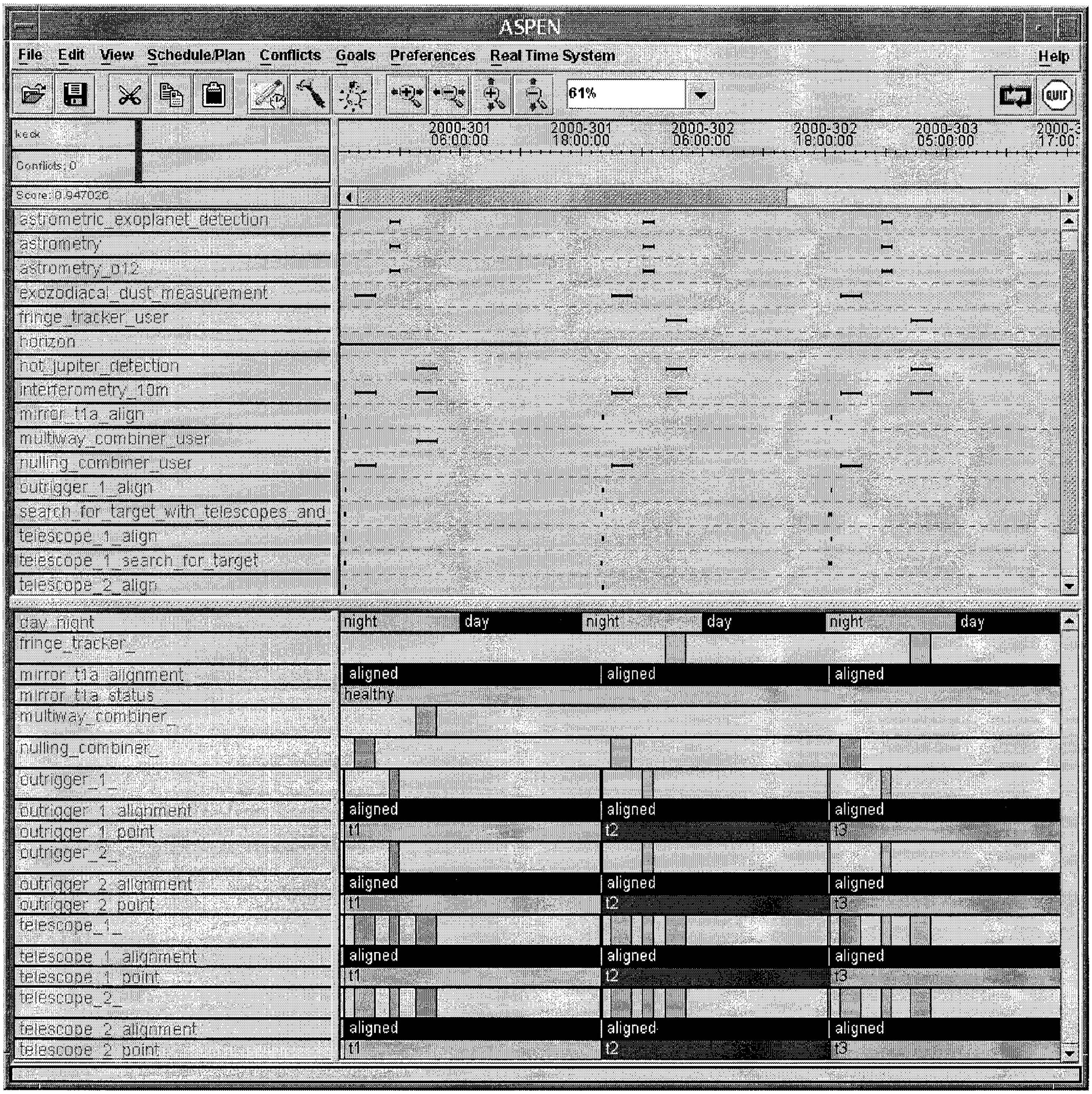



Figure 4 - ASPEN GUI for Keck 3-night observation schedule.

*KI Preference Model*

While the repaired plan satisfied all of the hard constraints, it would not be considered a high quality plan. In order to define plan quality, and subsequently optimize plans, we need to state a set of preferences for KI operations. First, the most common preference is one that prefers more science experiments. While the original request had nine experiments, it is possible that more could fit in the three-night plan. The user can make optional requests for additional science experiments. The corresponding preference will evaluate to a higher score when the plan contains more of the optional experiments. Other preferences are for smaller start times for activities. These make sure experiments are completed as soon as possible, within the requested start time window. Still more preferences are for fewer state changes on the pointing and alignment state variables. These keep the telescope and mirror movements at a minimum.

Now that we have the preference model defined, we can use the iterative optimization algorithm to improve the plan. Each preference is evaluated to a score between 0 and 1, with 1 being the highest score. These individual scores are then weighed and averaged into an overall score for the plan. The iterative optimization algorithm selects and improves one of the preferences until all have a score of one or a time limit has been exceeded. For example, it will delete unnecessary alignment activities to improve the score for the preference for fewer alignment state changes. It will also move activities to earlier start times (while avoiding conflicts) to improve other preferences.

*KI Continuous Planning Interface*

Finally, we have developed the required interface between the CASPER continuous planner and a KI simulator. Most of the code for the interface and sim were automatically generated from the ASPEN model. The generated sim is a discrete event simulator that simulates the execution of activities and the evolution of resources and state variables. The generated sim executes activities as expected by the planner. However, the user can augment the sim code to simulate variations at execution time. The generated interface links CASPER to the simulator. It is code that must match a specific template required by CASPER. In particular, it must have a function "`commit(Activity)`" that, given a CASPER activity, translates the activity into a simulator command. Because these commands are time tagged, the simulator will queue the commands and pop them off the queue at the appropriate time. In addition, the interface code translates the execution status into plan updates when the simulated state deviates from the planned state at any time. Whenever the plan updates result in conflicts, the repair algorithm is automatically invoked to fix them.

To understand this process, we will step through an example of a plan being executed by the simulator. First, we start ASPEN and generate a feasible plan. Then, we start CASPER, indicating the hostname and port number where the ASPEN socket server is running. This will allow CASPER to communicate with ASPEN. Next, we start the simulator and indicate the host and port where CASPER is running. This allows CASPER to communicate with the sim. As the simulator runs, CASPER monitors the current time reported by the sim. When the current time approaches the start time of a planned activity (within a predefined delta), CASPER reads the activity from ASPEN and calls the "`commit`" interface function. For example, one of the first planned activities for Keck is a "`search_for_target`" activity. One minute before the activity's start time, CASPER calls "`commit`" on this activity to translate it into a "`search_for_target`" simulator command with the target parameter value (e.g., "`t1`"). The simulator pushes this command in its queue. One minute later, the simulator pops it off the queue and begins simulating the execution of this command. For this command, the simulator simply waits for a specified duration, then changes a simulator variable to a new target value. If this value does not match the planned telescope pointing value, the plan will be updated to the actual (simulated) value. For example, if the command failed, it may change the variable to the value "`unknown`." This will likely cause a conflict because the experiments will require the telescope to be pointing at a particular target. CASPER will start the repair algorithm to fix any conflicts. In this example, one option for repair might be to use one of the other telescopes (if one continues to fail). As the execution simulation proceeds, CASPER continues to submit planned activities.

For all command types, with one exception, the CASPER interface for KI submits commands to a simulator. However, using hardware and software from the Keck Interferometer Testbed at JPL, we pass mirror alignment commands (for an arbitrary mirror along one of the beam trains) through an environment similar to the eventual KI environment. These commands are not simulated, but instead are sent to a mirror controller. In addition, the controller is queried for values of variables affected by these commands, such as the alignment status and health. The two-way communication with the controller is implemented using the Channel Access component of the EPICS software system. The Channel Access library provides a client/server interface to process variables in the EPICS database built for the control system. For this part of the CASPER interface, commands are sent as requests that change process variables, and plan updates are determined from requests that query variables.

## 8. Conclusions

We have demonstrated the utility of AI planning and scheduling technology for interferometer configuration and control. Interferometers are complex systems that require automation at many levels. We have shown that high-level AI planning can be used in conjunction with low-level sequencing to increase automation of these systems.

## Acknowledgment

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

## References

[1] M. Colavitaa, A. Bodena, S. Crawforda, A. Meinela, M. Shaoa, P. Swansona, G. van Bellea, G. Vasishta, J. Walkerb, J. Wallacea, P. Wizinowichb, "The Keck Interferometer," in Astronomical Interferometry, Proc. SPIE 3350 (1998).

[2] L. Dalesio, M. Kraimer, A. Kozubal, "EPICS Architecture," in ICALEPCS 1991, pp.278-281.

[3] A. Kozubal, "State Notation Language and Sequencer Users Guide," Los Alamos National Laboratory.

[4] S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, D. Tran, "ASPEN – Automating Space Mission Operations using Automated Planning and Scheduling," SpaceOps 2000, Toulouse, France, 2000.

[5] A. Fukunaga, G. Rabideau, S. Chien, D. Yan, "Toward an Application Framework for Automated Planning and Scheduling," Proc. Intl. Symp. of Artificial Intelligence, Robotics & Automation for Space, Tokyo, Japan, July 1997.

[6] F. Fisher, R. Knight, B. Engelhardt, S. Chien, N. Alejandre, "A Planning Approach to Monitor and Control for Deep Space Communications", Proceedings of the IEEE Aerospace Conference (IAC), Big Sky, MT, March 2000.

[7] R. Sherwood, A. Govindjee, D. Yan, G. Rabideau, S. Chien, A. Fukunaga, "Using ASPEN to Automate EO-1 Activity Planning," Proceedings of the 1998 IEEE Aerospace Conference, Aspen, CO, March 1998.

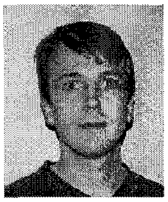
[8] M. Zweben, B. Daun, E. Davis, and M. Deale, "Scheduling and Rescheduling with Iterative Repair," in Intelligent Scheduling, Morgan Kaufman, San Francisco, 1994.

[9] G. Rabideau, R. Knight, S. Chien, A. Fukunaga, A. Govindjee, "Iterative Repair Planning for Spacecraft Operations in the ASPEN System," International Symposium on Artificial Intelligence Robotics and Automation in Space, Noordwijk, The Netherlands, June 1999.

[10] A. Jonsson, P. Morris, N. Muscettola, K. Rajan, and B. Smith, "Planning in Interplanetary Space: Theory and Practice," Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems, Breckenridge, CO, April 2000.

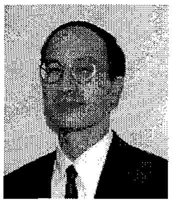
[11] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "Using Iterative Repair to Improve Responsiveness of Planning and Scheduling," Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling, Breckenridge, CO, April 2000.

[12] G. Rabideau, B. Engelhardt, S. Chien, "Using Generic Preferences to Incrementally Improve Plan Quality," Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling, Breckenridge, CO. April, 2000.



***Gregg Rabideau*** *is a senior member of the Artificial Intelligence Group at JPL where he has developed several automated planning systems for spacecraft and rover applications. He is the Software Lead for the ASPEN planning system, which received honorable mention for the 1999 NASA Software of the Year award. Gregg was lead for the Data-Chaser shuttle payload planner and for the planning search component of the Remote Agent. The Remote Agent was co-winner of the same 1999 NASA Software of the Year award. More recently, he has been leading group efforts in plan optimization. He holds B.S. and M.S. degrees in Computer Science at the University of Illinois.*

***Leonard Reder*** *is a senior member of the Real Time Control group within the Interferometry Systems and Technology Section at JPL. He is currently involved in development of sequencing software for the implementation of the Keck Interferometer. Prior to joining JPL he worked on a variety of software and hardware projects involving remote sensing for the U. S. Navy. Prior to this he was a hardware designer with Symbolics Inc. implementing graphics hardware as part of the companies Lisp based graphics workstation product. He holds a MSEE degree from the University of Southern California and a B.S. in Electronic Engineering from Cal Poly University at San Luis Obispo.*



***Steve Chien*** *is Technical Group Supervisor of the Artificial Intelligence Group of the Jet Propulsion Laboratory, California Institute of Technology where* *he leads efforts in research and development of automated planning and scheduling systems for science data analysis, ground station automation, and highly autonomous spacecraft. He is also on the Faculty of the Department of Computer Science at the University of Southern California. He holds a B.S., M.S., and Ph.D. in Computer Science from the University of Illinois. His research interests are in the areas of: planning and scheduling, operations research, and machine learning and he has published numerous articles in these areas. In 1995 he received the Lew Allen Award for Excellence and in 1997 he received a NASA Exceptional Engineering Achievement Medal both for his research and engineering work in automated planning and scheduling systems.*

***Andrew Booth** is the real time control system lead for the Keck Interferometer Project, part of the Interferometry Systems and Technology Section at JPL, and part of the NASA Origins Program. Prior to joining JPL he was a faculty member in the School of Physics at Sydney University, Australia, where he worked on the Sydney University Stellar Interferometer project. He holds a BA in physics and a DPhil in astrophysics from Oxford University, England. His research interests are in basic stellar astrophysics and building optical stellar interferometers, and he has published extensively in both areas.*